

10/21/2003 1:12 PM

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The program execution equipment which is program execution equipment which performs the garbage collection which collects the used fields included in program execution at a dynamic storage region, and makes into a free space, and is characterized by to have the real way stage which performs a garbage collection according to a surveillance means supervise the situation of a free space in a dynamic storage region, a maintenance means hold the limit information which restricts execution of a garbage collection, and the aforementioned situation and the aforementioned limit information.

[Claim 2] It is program execution equipment according to claim 1 carry out performing a garbage collection when the aforementioned limit information is the information which specifies prohibition or permission of execution of a garbage collection, the aforementioned surveillance means supervises the rate of the free space occupied in a dynamic storage region, the aforementioned rate becomes [means / execution / aforementioned] below in a predetermined value and the limit information currently held at the aforementioned maintenance means is the information specify the aforementioned permission as the feature.

[Claim 3] It is program execution equipment according to claim 2 which the information which specifies the aforementioned permission specifies the upper limit of the execution time of 1 time of a garbage collection, and is characterized by the aforementioned execution means performing the execution-time part garbage collection when the aforementioned rate becomes below a predetermined value and the limit information currently held at the aforementioned maintenance means has specified the upper limit of the execution time.

[Claim 4] It is program execution equipment according to claim 2 which the information which specifies the aforementioned permission specifies the capacity of the free space which should be secured in 1 time of a garbage collection, and is characterized by for the aforementioned execution means to perform the capacitive component garbage collection of the aforementioned free space when the aforementioned rate becomes below a predetermined value and the limit information currently held at the aforementioned maintenance means has specified the capacity of the aforementioned free space.

[Claim 5] It is program execution equipment according to claim 2 to 4 which the aforementioned limit information is described at the head of each portion according to the need for the real-time operation in each portion which constitutes a program, and is characterized by the aforementioned maintenance means holding it from the time of the aforementioned control information being read from the aforementioned program.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] this invention relates to program execution equipment equipped with a garbage collector.

[0002]

[Description of the Prior Art] The storage region which program execution equipment uses at the time of program execution is roughly divided into the following fields. (1) The field assigned to a code, a constant, etc. from the start of a program before an end. (2) The field assigned to storage, such as global-area data and a static data, from the start of a program before an end. (3) The field which is assigned in connection with program execution, or is released.

[0003] Furthermore, the field of (3) is divided into two fields. That is, they are the field assigned to a local parameter and the field assigned to dynamic data storages. These two fields are called the stack area and heap area from the data structure, respectively. At first, two or more free space was connected by the link, and the heap area has accomplished the list structure. In assigning a field to dynamic data, the field for the need is started from the free space of sufficient size under list, and it stores dynamic data. The field where this dynamic data was stored is called field during use. The dynamic data stored in the field during use are the process of program execution, and somewhere may not also no longer be referred to. Thus, when somewhere are also no longer referred to for dynamic data, a field is called used field during the use. This used field is again returned to a free space, and is connected with the list of free space.

[0004] Since the heap area is limited, it is consumed in connection with program execution, and the free space in a heap area decreases in number. If free space runs short, program execution equipment must change a used field into the situation where the newly generated dynamic data are unstorable, by releasing a used field by *****'s in a free space, before free space runs short. This processing is called garbage collection (it abbreviates to GC hereafter).

[0005] In the program execution equipment of processors, such as C and C++, the directions which perform GC are left to the programmer. That is, a programmer needs to specify the instruction (release instruction) which directs release of a used field into a program in a suitable part. However, since a programmer must describe a release instruction in detail in this case, it becomes the burden of a programmer. Moreover, a programmer cannot describe a release instruction and causes a critical error in that case.

[0006] On the other hand, the program execution equipment of processors, such as Java, is equipped with the garbage collector which performs GC that description of a release instruction is unnecessary, and automatically for the purpose of mitigating the burden concerning a programmer. If a garbage collector runs short of the free space in a heap area, GC will be performed automatically. The processing which a garbage collector performs is roughly divided into the following two stages. ** Judgment; classify data other than a free space in a heap area to a field and a used field during use. In here, judgment is based on whether it is possible to refer to directly [the field / in data areas, such as a register and a stack, to a pointer] or indirectly. ** Recovery; summarize two or more used fields which became small for reuse of a used field, and make a big free space.

[0007] By this garbage collector, since it becomes unnecessary for a programmer to pay attention entirely about GC, the burden of a programmer is mitigated.

[0008]

[Problem(s) to be Solved by the Invention] However, program execution equipment equipped with a garbage collector had the fault of not being suitable in the opposite side of the advantage which mitigates the burden of a programmer, and execution of a real-time operation. The reason is as follows. It is required that a real-time operation should end predetermined processing to within a time [fixed]. ***** [to which timing time execution of the GC by the garbage collector is carried out to it] of which. This timing is a time of a garbage collector detecting shortage of a free space, and is because the time concerning execution also changes with operating conditions of a heap area. Therefore, program execution equipment equipped with a garbage collector cannot predict time until predetermined processing is completed to within a time [fixed].

[0009] this invention aims at offering the program execution equipment suitable for execution of a real-time operation which performs a garbage collection in view of the above-mentioned problem, mitigating the burden concerning a programmer.

[0010]

[Means for Solving the Problem] In order to solve the above-mentioned problem, it is program execution equipment which

performs the garbage collection which the program execution equipment of this invention collects the used fields included in program execution at a dynamic storage region, and makes into a free space, and it has the real way stage which performs a garbage collection according to a surveillance means supervise the situation of a free space in a dynamic storage region, a maintenance means hold the limit information which restricts execution of a garbage collection, and the aforementioned situation and the aforementioned limit information.

[0011] Moreover, the aforementioned limit information is the information which specifies prohibition or permission of execution of a garbage collection, the aforementioned surveillance means supervises the rate of the free space which occupies in a dynamic storage region, and when the aforementioned rate becomes below in a predetermined value and the limit information currently held at the aforementioned maintenance means is the information specify the aforementioned permission, the aforementioned execution means is constituted so that a garbage collection may perform.

[0012] Moreover, the information which specifies the aforementioned permission specifies the upper limit of the execution time of 1 time of a garbage collection, and when the aforementioned rate becomes below a predetermined value and the limit information currently held at the aforementioned maintenance means has specified the upper limit of the execution time, the aforementioned execution means is constituted so that the execution-time part garbage collection may be performed. Moreover, the information which specifies the aforementioned permission specifies the capacity of the free space which should be secured in 1 time of a garbage collection, and when the aforementioned rate becomes below a predetermined value and the limit information currently held at the aforementioned maintenance means has specified the capacity of the aforementioned free space, the aforementioned execution means is constituted so that the capacitive component garbage collection of the aforementioned free space may be performed.

[0013] Furthermore, the aforementioned limit information is described at the head of each portion according to the need for the real-time operation in each portion which constitutes a program, and the aforementioned maintenance means is constituted so that it may be held from the time of the aforementioned control information being read from the aforementioned program.

[0014]

[Embodiments of the Invention] The program execution equipment applied to the operation gestalt of this invention below is explained using a drawing. This program execution equipment is constituted so that execution of GC may be controlled in parallel to the program execution inputted according to the limit information about GC execution described in the program, and the situation of the free space of a heap area.

[0015] Drawing 1 is the block diagram showing the composition of program execution equipment. In this drawing, program execution equipment 1 consists of the program storing section 3, the storage section 4, the instruction-processing section 5, the renewal section 6 of a flag, the flag attaching part 7, the rolling-counters-forward section 8, the counter value attaching part 9, the field Monitoring Department 10, the GC starting section 11, and the GC processing section 12. The program storing section 3 memorizes the program inputted from the exterior. As for the program, in addition to the byte code, limit information is described in here. the instruction to which limit information restricts the execution of GC in a program -- it is -- three kinds, a time-limit instruction, a restraining order, and a permission instruction, -- it is .

[0016] An example is used for below and these limit information is explained to it. Drawing 2 shows an example of the program limit information was described to be. In this drawing, gc#limit() in Instruction A and Instruction D is a time-limit instruction. This instruction restricts the execution time of GC performed at once. Suppose that the execution time restricted by this instruction is called time limit. The concrete time limit is described in the parenthesis. For example, in gc#limit(200), the time limit is 200ms. The section where this instruction is effective is from the line this instruction is described to be to the line limit information is described to be below. For example, in the period when the byte code described between section A-B of this drawing is performed, program execution equipment 1 ends GC performed at once within 200ms. After program execution equipment 1 started execution of GC in here, when 200ms has passed, even if processing is not completed, it is once interrupted, and the remaining processing is turned to the following opportunity.

[0017] gc#forbid() in Instruction B is a restraining order. A restraining order forbids GC execution. The section where this instruction is effective is like a time-limit instruction from the line this instruction is described to be to the line limit information is described to be below. For example, in the period when the byte code described between section B-C of this drawing is performed, execution of GC is forbidden and program execution equipment 1 does not perform GC.

[0018] gc#permit() in Instruction C is a permission instruction. This instruction permits GC execution. In other words, it is the instruction which cancels a limit of execution by a time-limit instruction or restraining order. The section where this instruction is effective is like a time-limit instruction from the line this instruction is described to be to the line limit information is described to be below. For example, in the period when the byte code described between section C-D of this drawing is performed, program execution equipment 1 performs GC, when required.

[0019] A programmer describes in a program the limit information shown above as a restraining order is described in the direct continued line of the instruction group in which a real-time operation is made to perform and a permission instruction is described in the direct continued line of the instruction group which does not perform a real-time operation. Since this should just specify prohibition of execution of GC, the time limit, and the range of permission to the program execution equipment of the conventional C or the processor of C++ having had to describe allocation and release of a field in detail, the burden concerning a programmer is mitigated.

[0020] The storage section 4 consists of a stack area and a heap area. The instruction-processing section 5 performs allocation of these fields. The instruction-processing section 5 reads and executes [decode and] 1 **** of instructions at a time from

the program storing section 3, and performs allocation to the dynamic data of a heap area if needed. The renewal section 6 of a flag updates the value of the permission flag which the flag attaching part 7 holds, when the instruction read by the instruction-processing section 5 is a restraining order or a permission instruction. More specifically, when it is a permission instruction, a value is set to 1, and when it is a restraining order, a value is reset to 0. Moreover, the renewal section 6 of a flag outputs the signal S1 which shows the purport which updated the value to GC starting section 11, when a value is updated.

[0021] The flag attaching part 7 has a permission flag, and holds the flag value set / reset by the renewal section 6 of a flag. The rolling-counters-forward section 8 updates the time limit which the counter value attaching part 9 holds, when the instruction read by the instruction-processing section 5 is a time-limit instruction. Moreover, the rolling-counters-forward section 8 outputs the signal S2 which shows the purport which updated the time limit to GC starting section 11, when the time limit is updated.

[0022] The counter value attaching part 9 holds the time limit updated by the rolling-counters-forward section 8. Drawing 3 is the flow chart of the procedure consisting mainly of the instruction-processing section 5, the renewal section 6 of a flag, and the rolling-counters-forward section 8. If a program is first stored in the program storing section 3 in this drawing, the instruction-processing section 5 will process the preceding paragraph story of instruction decodes, such as reservation of the storage section, and initialization of a variable, (Step 301, Step 302).

[0023] from the program storing section 3, 1 instruction reading appearance of the instruction-processing section 5 is carried out, and it is decoded. As a result of decoding, when an instruction is a restraining order or a permission instruction, the renewal section 6 of a flag updates the value of the permission flag of the flag attaching part 7, and outputs a signal S1 to GC starting section 11 (Step 304, Step 305). As a result of decoding, when an instruction is a time-limit instruction, the rolling-counters-forward section 8 updates the counter value of the counter value attaching part 9, and outputs a signal S2 to GC starting section 11 simultaneously (Step 306, Step 307).

[0024] In not being which limit information as a result of decode, either, the instruction-processing section 5 executes the instruction decoded to usual (Step 308). Thus, after a program is completed (Step 309) (i.e., if all instructions are read from the program storing section 3 and the renewal of a flag, the renewal of a counter value, or execution of an instruction is completed), the instruction-processing section 5 ends processing, after performing after treatment, such as release of the storage section, (Step 310).

[0025] Decode of an instruction, renewal of a flag, and renewal of the time limit are performed [in / the instruction-processing section 5, the renewal section 6 of a flag, and the rolling-counters-forward section 8 / as mentioned above]. The field Monitoring Department 10 supervises the situation of the free space of a heap area. Specifically, the field Monitoring Department 10 holds the address of the free space which adjoins the backside of a field during the use, when a free space divided and shines to dynamic data and turns into a field during use. It is shown that a free space in a heap area is decreasing, so that the address of this free space is close to the address of a tail. This is because it is assigned to dynamic data sequentially from a free space in the head of a heap area. The address which the field Monitoring Department 10 holds is used for GC starting section 11.

[0026] GC starting section 11 controls execution of GC according to the present limit information and the situation of a free space. In detail, when it judges with GC starting section 11 judging whether free space is insufficient from the address of the free space held once (for example, 1 time per decode execution of four instructions) at the field Monitoring Department 10, and running short it, whenever decode execution of the instruction for a predetermined number is carried out by the instruction-processing section 5, according to the present limit information, GC execution by GC processing section 12 is controlled.

[0027] Moreover, GC starting section 11 has the timer (not shown), when a time-limit instruction starts GC to the effective bottom, it starts a timer, and it terminates GC within the time limit. GC starting section 11 has an update flag, in order to grasp what the present limit information is. It is shown here which [which holds the more nearly newest content among the content of the flag attaching part 7 and the content of the counter attaching part 9 and which that is, / effective / an update flag / now] limit information with is held. GC starting section 11 sets an update flag to 1, when a signal S1 is outputted from the renewal section 6 of a flag, and when a signal S2 is outputted from the rolling-counters-forward section 8, it resets an update flag to 0. That is, it is shown that the permission instruction or restraining order held at the flag attaching part 7 is effective if an update flag is 1, a time-limit instruction is effective if an update flag is 0, and the time limit is held at the counter value attaching part 9.

[0028] Drawing 4 is a flow chart which shows the procedure of GC control processing by GC starting section 11. GC starting section 11 performs processing shown in this flow chart once (for example, decode of four instructions, and 1 time per execution) whenever the instruction for a predetermined number is decoded and executed by the instruction-processing section 5. GC starting section 11 judges whether the rate of the size of the free space occupied to a heap area is 1/2 or less from the address of the free space held at the field Monitoring Department 10 (Step 401).

[0029] When it judges with it being 1/2 or less in Step 401, as for GC starting section 11, an update flag judges 1 or 0 (Step 402). When it judges with an update flag being 1 in Step 402, it judges whether GC starting section 11 is [0] whether the value of the permission flag held at the flag attaching part 7 is 1 (Step 403).

[0030] When it judges with a permission flag being 1 in Step 403, GC starting section 11 makes GC processing section 12 perform GC (Step 404). When it judges with a permission flag being 0 in Step 403, GC starting section 11 ends processing of this flow chart. On the other hand, when it judges with an update flag being 0 in Step 402, GC starting section 11 sets a timer

to the value of the time limit currently held at the counter value attaching part 9, and makes a timer count start (Step 405). [0031] Next, GC starting section 11 makes GC processing section 12 perform GC (Step 406). GC starting section 11 will stop the execution of GC by GC processing section 12, if the time limit passes by the timer count (Step 407). GC in Step 406 is interrupted if processing is not completed when the time limit passes. Interrupted GC is performed from the place interrupted at the time of next GC execution.

[0032] When it judges with GC starting section 11 running short of free space as mentioned above as for program execution equipment 1, GC processing section 12 is controlled according to the limit information at that time. That is, it is the control which performs GC and which does not perform GC that less than the time limit performs GC. Thus, since program execution equipment 1 performs GC appropriately according to the limit information and situation of a free space only only by appointing the section which permits execution of GC by constituting this program execution equipment 1, the section which permit with the time limit, and the section forbid, and describing on a program, it is effective in the burden which starts a programmer compared with the case where a release instruction is described in detail like before being mitigated.

[0033] Moreover, if it describes that a restraining order becomes effective in the section of the real-time operation on a program, since the section does not perform GC, program execution equipment 1 has the effect perform a real-time operation and that things can be carried out in program execution equipment 1, without being influenced of GC. Although the program execution equipment applied to this invention above was explained based on the operation gestalt, you may carry out this invention not only in these execution gestalten but as follows.

(1) Although the rate of the size of the free space which occupies GC starting section 11 to a heap area has judged whether it is 1/2 in Step 401, 1/3, 1/4, 3/5, etc. are good anything, for example. If it takes using a field efficiently into consideration, as for this value, it is desirable that it is or less about 1 / 2 from zero or more.

In Step 403 (2) GC starting section 11 It judges whether when it judges with a permission flag being 0, the rate of the size of the free space shown in a heap area from the address of the free space held further at the field Monitoring Department 10 is 1/4 (this value should be just smaller than the value set up in Step 401) or less. When it judges or less with 1/4, GC processing section 12 is made to perform GC, and when it judges with it not being 1/4 or less, you may constitute so that processing of this flow chart may be ended.

(3) Instead of setting up the time limit of a time-limit instruction, you may make it set up the capacity of the free space which should be secured in 1 time of GC. That is, in 1 time of GC, you may constitute so that it may say that the GC is ended when the capacity returned to a free space from the used field reaches the specified quantity (for example, 1/4 amount of a heap area).

(4) The limit information which carries out composition of the program execution equipment 1 shown in indrawing 1 to the composition except the rolling-counters-forward section 8 and the counter value attaching part 9, and can describe a programmer is good only as for the restraining order and permission instruction except a time-limit instruction.

[0034]

[Effect of the Invention] The program execution equipment of this invention is equipped with the real way stage which is program execution equipment which performs the garbage collection which collects the used fields included in program execution at a dynamic storage region, and is made into a free space, and performs a garbage collection according to a surveillance means supervise the situation of a free space in a dynamic storage region, a maintenance means hold the limit information which restricts execution of a garbage collection, and the aforementioned situation and the aforementioned limit information.

[0035] According to this composition, since the real way stage performs a garbage collection according to the situation and limit information on a free space, a programmer does not need to describe a release instruction in detail in order to make a garbage collection perform, and is effective in the burden concerning a programmer being mitigated. Moreover, since ending predetermined processing to within a time [fixed] is guaranteed compared with the garbage collection according like before to the garbage collector [***** / timing and the execution time / the collector] since the garbage collection by the real way stage is restricted by limit information, this program execution equipment is suitable with execution of a real-time operation.

[0036] Moreover, the aforementioned limit information is the information which specifies prohibition or permission of execution of a garbage collection, the aforementioned surveillance means supervises the rate of the free space which occupies in a dynamic storage region, and when the aforementioned rate becomes below in a predetermined value and the limit information currently held at the aforementioned maintenance means is the information specify the aforementioned permission, the aforementioned execution means is constituted so that a garbage collection may perform.

[0037] According to this composition, when a real-time operation is required in a program, a garbage collection is forbidden, and a garbage collection can be permitted when not required. Moreover, since the real way stage performs a garbage collection in the section where the information which specifies the aforementioned permission on a program is effective when the aforementioned rate becomes below a predetermined value, the burden which it becomes unnecessary for a programmer to direct execution of a garbage collection in detail, and starts a programmer is mitigated.

[0038] Moreover, the information which specifies the aforementioned permission is characterized by specifying the upper limit of the execution time of 1 time of a garbage collection. Since the real way stage can restrict by this the execution time of the garbage collection performed at once, the time which processing of a program takes can be predicted. Therefore, this program execution equipment can perform the real-time operation program as which predictability is required as

requirements.

[0039] Moreover, the information which specifies the aforementioned permission is characterized by specifying the capacity of the free space which should be secured in 1 time of a garbage collection. Thereby, since it will be interrupted if the garbage collection of a capacitive component is performed when a used field is larger than the aforementioned capacity, the real way stage can hold down the time concerning 1 time of a garbage collection to a capacitive component. Therefore, since it can make within a time [to which processing of a program is expected] completed, this program execution equipment fits execution of a real-time operation.

[0040] Moreover, the aforementioned limit information is described at the head of each portion according to the need for the real-time operation in each portion which constitutes a program, and the aforementioned maintenance means is constituted so that it may be held from the time of the aforementioned control information being read from the aforementioned program. Since a programmer can describe the information which specifies the ban on the above at the head of the portion which makes a real-time operation perform to program execution equipment, the information which specifies the aforementioned permission at the head of the portion into which a real-time operation does not make perform can only describe and execution of the garbage collection by the real way stage after the description can restrict by this, a programmer has an effect of it becoming unnecessary that it is necessary not to describe execution of a garbage collection on a program in detail.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is drawing showing the composition of program execution equipment.

[Drawing 2] An example of the program limit information was described to be is shown.

[Drawing 3] It is the flow chart of the procedure consisting mainly of the instruction-processing section 5, the renewal section 6 of a flag, and the rolling-counters-forward section 8.

[Drawing 4] It is the flow chart which shows the procedure of GC control processing by GC starting section 11.

[Description of Notations]

3 Program Storing Section

4 Storage Section

5 Instruction-Processing Section

6 Renewal Section of Flag

7 Flag Attaching Part

8 Rolling-Counters-Forward Section

9 Counter Value Attaching Part

10 Field Monitoring Department

11 GC Starting Section

12 GC Processing Section

[Translation done.]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2001-14215

(P2001-14215A)

(43)公開日 平成13年1月19日(2001.1.19)

(51)Int.Cl.⁷

G 0 6 F 12/00

識別記号

5 9 1

F I

G 0 6 F 12/00

テ-マコ-ト*(参考)

5 9 1 5 B 0 6 0

審査請求 未請求 請求項の数5 O L (全 7 頁)

(21)出願番号

特願平11-188143

(22)出願日

平成11年7月1日(1999.7.1)

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 金丸 智一

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者 富永 宜輝

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(74)代理人 100090446

弁理士 中島 可朗 (外1名)

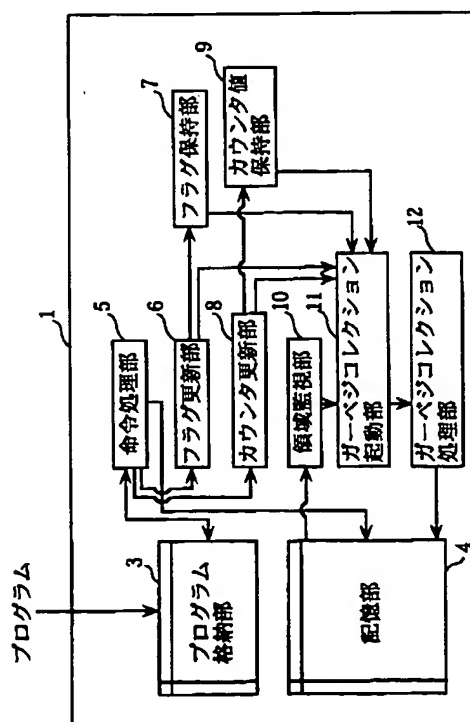
最終頁に続く

(54)【発明の名称】 プログラム実行装置

(57)【要約】

【課題】 プログラムにかかる負担を軽減しつつリアルタイム処理の実行に適したプログラム実行装置を提供する。

【解決手段】 プログラム実行装置1において、フラグ更新部6及びカウンタ更新部8は、プログラム中に記述されたガーベジコレクションの制限情報を保持する。領域監視部10は、記憶部4中のヒープ領域における未使用領域の状況を監視し、ガーベジコレクション起動部11は、保持されている前記制限情報と監視される前記状況とに応じてガーベジコレクション処理部12によるガーベジコレクションの実行を制御する。



【特許請求の範囲】

【請求項1】 プログラムの実行中に、動的記憶領域に含まれる使用済み領域を回収して未使用領域にするガーベジコレクションを行うプログラム実行装置であって、動的記憶領域における未使用領域の状況を監視する監視手段と、

ガーベジコレクションの実行を制限する制限情報を保持する保持手段と、

前記状況と前記制限情報とに応じてガーベジコレクションを実行する実行手段とを備えることを特徴とするプログラム実行装置。

【請求項2】 前記制限情報は、ガーベジコレクションの実行の禁止又は許可を規定する情報であり、

前記監視手段は、動的記憶領域中に占める未使用領域の割合を監視し、

前記実行手段は、前記割合が所定値以下になった場合に、前記保持手段に保持されている制限情報が前記許可を規定する情報であるときガーベジコレクションを実行することを特徴とする請求項1記載のプログラム実行装置。

【請求項3】 前記許可を規定する情報は、1回のガーベジコレクションの実行時間の上限を規定し、

前記実行手段は、前記割合が所定値以下になった場合に、前記保持手段に保持されている制限情報が実行時間の上限を規定しているとき、その実行時間分ガーベジコレクションを実行することを特徴とする請求項2記載のプログラム実行装置。

【請求項4】 前記許可を規定する情報は、1回のガーベジコレクションにおいて確保すべき未使用領域の容量を規定し、

前記実行手段は、前記割合が所定値以下になった場合に、前記保持手段に保持されている制限情報が前記未使用領域の容量を規定しているとき、その前記未使用領域の容量分ガーベジコレクションを実行することを特徴とする請求項2記載のプログラム実行装置。

【請求項5】 前記制限情報は、プログラムを構成する各部分におけるリアルタイム処理の必要性に応じて各部分の先頭に記述され、

前記保持手段は、前記プログラムより前記制御情報が読み出された時点からそれを保持することを特徴とする請求項2乃至4のいずれかに記載のプログラム実行装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ガーベジコレクタを備えるプログラム実行装置に関する。

【0002】

【従来の技術】 プログラム実行装置がプログラムの実行時に使用する記憶領域は、大きく以下の領域に分けられる。(1)プログラムの開始から終了までの間、コードや定数等に割り当てられる領域。(2)プログラムの開

始から終了までの間、大域データや静的データ等の記憶用に割り当てられる領域。(3)プログラムの実行に伴って割り当てられたり解放されたりする領域。

【0003】さらに(3)の領域は2つの領域に分けられる。すなわち局所変数に割り当てられる領域と動的データの記憶用に割り当てられる領域である。この2つの領域は、そのデータ構造からそれぞれスタック領域とヒープ領域と呼ばれている。ヒープ領域は、最初、複数の未使用領域がリンクでつながれてリスト構造を成している。動的データに対して領域の割当てを行う場合には、リスト中の十分な大きさの未使用領域から必要分の領域を切り出して動的データを格納する。この動的データが格納された領域は、使用中領域と呼ばれる。使用中領域に格納されている動的データは、プログラム実行の過程で、どこからも参照されなくなることがある。このように動的データがどこからも参照されなくなったとき、その使用中領域は使用済み領域と呼ばれる。この使用済み領域は、再び未使用領域に戻されて未使用領域のリストにつながる。

【0004】ヒープ領域は有限であるから、プログラムの実行に伴って消費されてゆき、ヒープ領域中の未使用領域が減少していく。未使用領域が不足すると、新たに発生した動的データが格納できない事態に陥いるので、プログラム実行装置は、未使用領域が不足する前に使用済み領域を解放することによって、使用済み領域を未使用領域に変更しなければならない。この処理をガーベジコレクション(以下、GCと省略する)と呼ぶ。

【0005】CやC++等の処理系のプログラム実行装置においては、GCを実行させる指示をプログラマに任せている。つまりプログラマは、プログラム中に使用済み領域の解放を指示する命令(解放命令)を適切な箇所に明記する必要がある。しかしこの場合、プログラマは解放命令を逐一記述しなければならないので、プログラマの負担になる。またプログラマは解放命令を記述し損ねることがあり、その場合は重篤なエラーの原因となる。

【0006】これに対してJava等の処理系のプログラム実行装置は、プログラマにかかる負担を軽減することを目的として、解放命令の記述の必要なく自動的にGCを行うガーベジコレクタを備える。ガーベジコレクタは、ヒープ領域中の未使用領域が不足すると、自動的にGCを実行する。ガーベジコレクタが行う処理は、大きく次の2段階に分かれる。①分別；ヒープ領域中の未使用領域以外のデータを使用中領域と使用済み領域とに分別する。ここにおいて分別は、その領域がレジスタやスタックなどのデータ領域からポインタで直接あるいは間接に参照することが可能か否かによる。②回収；使用済み領域の再利用のため、小さくなった複数の使用済み領域をまとめて大きな未使用領域を作り出す。

【0007】このガーベジコレクタにより、プログラマ

はGCに関して一切注意を払う必要がなくなるのでプログラムの負担が軽減される。

【0008】

【発明が解決しようとする課題】ところがガーベジコレクタを備えるプログラム実行装置は、プログラムの負担を軽減する利点の反面、リアルタイム処理の実行には適さないという欠点があった。その理由は以下の通りである。リアルタイム処理は一定の時間内に所定の処理を終了することが要求される。それに対してガーベジコレクタによるGCは、どのタイミングでどれだけの時間実行されるかが予測不可能である。というのはこのタイミングは、ガーベジコレクタが未使用領域の不足を検出したときであり、また実行にかかる時間も、ヒープ領域の使用状況によって異なるからである。よってガーベジコレクタを備えるプログラム実行装置は、一定の時間内に所定の処理が終了するまでの時間を予測できないのである。

【0009】上記問題に鑑みて本発明は、プログラムにかかる負担を軽減しつつリアルタイム処理の実行に適した、ガーベジコレクションを行うプログラム実行装置を提供することを目的とする。

【0010】

【課題を解決するための手段】上記問題を解決するため、本発明のプログラム実行装置は、プログラムの実行中に、動的記憶領域に含まれる使用済み領域を回収して未使用領域にするガーベジコレクションを行うプログラム実行装置であって、動的記憶領域における未使用領域の状況を監視する監視手段と、ガーベジコレクションの実行を制限する制限情報を保持する保持手段と、前記状況と前記制限情報とに応じてガーベジコレクションを実行する実行手段とを備える。

【0011】また前記制限情報は、ガーベジコレクションの実行の禁止又は許可を規定する情報であり、前記監視手段は、動的記憶領域中に占める未使用領域の割合を監視し、前記実行手段は、前記割合が所定値以下になった場合に、前記保持手段に保持されている制限情報が前記許可を規定する情報であるときガーベジコレクションを実行するよう構成される。

【0012】また前記許可を規定する情報は、1回のガーベジコレクションの実行時間の上限を規定し、前記実行手段は、前記割合が所定値以下になった場合に、前記保持手段に保持されている制限情報が実行時間の上限を規定しているとき、その実行時間分ガーベジコレクションを実行するよう構成される。また前記許可を規定する情報は、1回のガーベジコレクションにおいて確保すべき未使用領域の容量を規定し、前記実行手段は、前記割合が所定値以下になった場合に、前記保持手段に保持されている制限情報が前記未使用領域の容量を規定しているとき、その前記未使用領域の容量分ガーベジコレクションを実行するよう構成される。

【0013】さらに前記制限情報は、プログラムを構成する各部分におけるリアルタイム処理の必要性に応じて各部分の先頭に記述され、前記保持手段は、前記プログラムより前記制御情報が読み出された時点からそれを保持するよう構成される。

【0014】

【発明の実施の形態】以下本発明の実施形態にかかるプログラム実行装置について図面を用いて説明する。本プログラム実行装置は、入力されるプログラムの実行と並行して、そのプログラム中に記述されたGC実行に関する制限情報とヒープ領域の未使用領域の状況とに応じてGCの実行を制御するよう構成されている。

【0015】図1は、プログラム実行装置の構成を示すブロック図である。同図においてプログラム実行装置1は、プログラム格納部3、記憶部4、命令処理部5、フラグ更新部6、フラグ保持部7、カウンタ更新部8、カウンタ値保持部9、領域監視部10、GC起動部11、GC処理部12から構成される。プログラム格納部3は、外部より入力されるプログラムを記憶する。ここにおいてプログラムは、バイトコードに加えて、制限情報が記述されている。制限情報は、プログラム中におけるGCの実行を制限する命令であり、時間制限命令、禁止命令、許可命令の3種類ある。

【0016】以下にこれらの制限情報を例を用いて説明する。図2は、制限情報が記述されたプログラムの一例を示す。同図において命令A及び命令Dにおけるgc#limit()は、時間制限命令である。この命令は、1回に実行されるGCの実行時間を制限する。この命令によって制限される実行時間を制限時間と呼ぶこととする。具体的な制限時間は括弧内に記述されている。例えばgc#limit(200)の場合、制限時間は200msである。この命令が有効な区間は、この命令が記述されている行から次に制限情報が記述されている行までである。例えば同図の区間A-Bの間に記述されているバイトコードが実行される期間においては、プログラム実行装置1は1回に実行するGCを200ms以内に終了する。ここにおいてプログラム実行装置1は、GCの実行を開始してから200msが経過した時点で、処理が完了していなくても一旦中断して、残りの処理を次の機会にまわす。

【0017】命令Bにおけるgc#forbid()は、禁止命令である。禁止命令は、GC実行を禁止する。この命令が有効な区間は、時間制限命令と同様、この命令が記述されている行から次に制限情報が記述されている行までである。例えば同図の区間B-Cの間に記述されているバイトコードが実行されている期間においては、GCの実行は禁止され、プログラム実行装置1はGCを実行しない。

【0018】命令Cにおけるgc#permit()は、許可命令である。この命令は、GC実行を許可する。言い替えば、時間制限命令または禁止命令による実行の制限を解

除する命令である。この命令が有効な区間は、時間制限命令と同様、この命令が記述されている行から次に制限情報が記述されている行までである。例えば、同図の区間C-Dの間に記述されているバイトコードが実行される期間においては、プログラム実行装置1は必要な場合にGCを実行する。

【0019】プログラマは、例えばリアルタイム処理を行わせる命令群の直前の行において禁止命令を記述し、リアルタイム処理を行わない命令群の直前の行において許可命令を記述するというように、以上に示した制限情報をプログラム中に記述する。これは従来のCやC++の処理系のプログラム実行装置が、領域の割当てと解放とを逐一記述しなければならなかったのに対し、GCの実行の禁止、制限時間、許可の範囲を指定するだけでよいので、プログラマにかかる負担は軽減される。

【0020】記憶部4は、スタック領域及びヒープ領域から構成される。これらの領域の割当ては、命令処理部5が行う。命令処理部5は、プログラム格納部3より1命令ずつ命令を読み出して解釈、実行し、必要に応じてヒープ領域の動的データへの割当てを行う。フラグ更新部6は、命令処理部5により読み出された命令が、禁止命令又は許可命令である場合に、フラグ保持部7が保持する許可フラグの値を更新する。より具体的には、許可命令である場合、値を1にセットし、禁止命令である場合、値を0にリセットする。またフラグ更新部6は、値を更新した時、GC起動部11に値を更新した旨を示す信号S1を出力する。

【0021】フラグ保持部7は、許可フラグを有し、フラグ更新部6によりセット／リセットされるフラグ値を保持する。カウンタ更新部8は、命令処理部5により読み出された命令が、時間制限命令である場合にカウンタ値保持部9が保持する制限時間を更新する。またカウンタ更新部8は、制限時間を更新したとき、GC起動部11に制限時間を更新した旨を示す信号S2を出力する。

【0022】カウンタ値保持部9は、カウンタ更新部8により更新される制限時間を保持する。図3は、命令処理部5、フラグ更新部6、カウンタ更新部8を中心とする処理手順のフローチャートである。同図においてまず、プログラム格納部3にプログラムが格納されると、命令処理部5は、記憶部の確保や、変数の初期化等の命令解釈の前段階の処理を行う（ステップ301、ステップ302）。

【0023】命令処理部5は、プログラム格納部3から1命令読み出して、解釈する。解釈した結果、命令が禁止命令又は許可命令である場合、フラグ更新部6は、フラグ保持部7の許可フラグの値を更新し、GC起動部11に信号S1を出力する（ステップ304、ステップ305）。解釈した結果、命令が時間制限命令である場合、カウンタ更新部8は、カウンタ値保持部9のカウンタ値を更新し、同時にGC起動部11に信号S2を出力

する（ステップ306、ステップ307）。

【0024】解釈の結果、いずれの制限情報でもない場合には、命令処理部5は、通常に解釈した命令を実行する（ステップ308）。このようにしてプログラムが終了すると、つまりプログラム格納部3から全ての命令を読み出してフラグの更新またはカウンタ値の更新又は命令の実行が完了すると（ステップ309）、命令処理部5は、記憶部の解放等の後処理を行った後、処理を終了する（ステップ310）。

【0025】以上のようにして命令処理部5、フラグ更新部6、カウンタ更新部8においては、命令の解釈、フラグの更新、制限時間の更新が行われる。領域監視部10は、ヒープ領域の未使用領域の状況を監視する。具体的には、領域監視部10は、未使用領域が動的データに割り当てられて使用中領域となったとき、その使用中領域の後ろ側に隣接している未使用領域のアドレスを保持する。この未使用領域のアドレスが、末尾のアドレスに近いほどヒープ領域中の未使用領域が少なくなってきたことを示す。これはヒープ領域の先頭にある未使用領域から順に動的データに割り当てられていくからである。領域監視部10が保持しているアドレスは、GC起動部11に使用される。

【0026】GC起動部11は、現在の制限情報と、未使用領域の状況とに応じてGCの実行を制御する。詳しくは、GC起動部11は、命令処理部5によって所定数分の命令が解釈実行される毎に1回（例えば4つの命令の解釈実行毎に1回）、領域監視部10に保持される未使用領域のアドレスから未使用領域が不足しているか否かを判定し、不足していると判定した場合には、現在の制限情報に応じてGC処理部12によるGC実行を制御する。

【0027】またGC起動部11は、タイマー（図示しない）を有しており、時間制限命令が有効となっている下においてGCを起動する場合にはタイマーを起動して、制限時間以内にGCを終了させる。GC起動部11は、現在の制限情報が何であるかを把握しておくために更新フラグを有する。ここで更新フラグは、フラグ保持部7の内容とカウンタ保持部9の内容とのうち、どちらがより最新の内容を保持しているか、つまりどちらが現在有効な制限情報を保持しているか、を示す。GC起動部11は、フラグ更新部6より信号S1が出力されたとき更新フラグを1にセットし、カウンタ更新部8より信号S2が出力されたとき更新フラグを0にリセットする。つまり更新フラグが1であればフラグ保持部7に保持される許可命令または禁止命令が有効であるし、更新フラグが0であれば時間制限命令が有効であって、その制限時間はカウンタ値保持部9に保持されていることを示す。

【0028】図4は、GC起動部11によるGC制御処理の手順を示すフローチャートである。GC起動部11

は、命令処理部5によって所定数分の命令が解説及び実行されるたびに1回(例えば、4つの命令の解説及び実行毎に1回)このフローチャートに示す処理を実行する。GC起動部11は、領域監視部10に保持される未使用領域のアドレスから、ヒープ領域に占める未使用領域の大きさの割合が $1/2$ 以下であるか否かを判定する(ステップ401)。

【0029】ステップ401において $1/2$ 以下であると判定した場合、GC起動部11は更新フラグが1か0かを判定する(ステップ402)。ステップ402において更新フラグが1であると判定した場合、GC起動部11はフラグ保持部7に保持される許可フラグの値が1であるか0であるかを判定する(ステップ403)。

【0030】ステップ403において許可フラグが1であると判定した場合には、GC起動部11は、GC処理部12にGCを実行させる(ステップ404)。ステップ403において許可フラグが0であると判定した場合には、GC起動部11はこのフローチャートの処理を終了する。一方、ステップ402において更新フラグが0であると判定した場合、GC起動部11は、タイマーをカウンタ値保持部9に保持されている制限時間の値にセットしてタイマーカウントを開始させる(ステップ405)。

【0031】次にGC起動部11は、GC処理部12にGCを実行させる(ステップ406)。GC起動部11は、タイマーカウントによって制限時間が経過するとGC処理部12によるGCの実行を停止させる(ステップ407)。ステップ406におけるGCは、制限時間が経過したとき処理が完了していなければ中断される。中断されたGCは、次のGC実行時に中断したところから行われる。

【0032】以上のようにしてプログラム実行装置1は、GC起動部11が未使用領域が不足していると判定した場合に、そのときの制限情報に応じてGC処理部12を制御する。すなわちGCを実行させる、GCを実行させない、GCを制限時間以内だけ実行させる、という制御である。このように本プログラム実行装置1を構成することにより、GCの実行を許可する区間、制限時間付きで許可する区間、禁止する区間を定めてプログラム上に記述するだけで、プログラム実行装置1がその制限情報と未使用領域の状況とに応じて適切にGCを実行するので、従来のように逐一解放命令を記述する場合と比べてプログラマにかかる負担が軽減されるという効果がある。

【0033】またプログラム上のリアルタイム処理の区間において禁止命令が有効になるように記述すれば、プログラム実行装置1はその区間はGCを実行しないので、プログラム実行装置1に、GCの影響を受けることなくリアルタイム処理を実行させることができるという効果がある。以上本発明にかかるプログラム実行装置につ

いて実施形態に基づいて説明したが本発明はこれらの実行形態に限らず、以下のようにしてもよい。

(1) ステップ401においてGC起動部11は、ヒープ領域に占める未使用領域の大きさの割合が $1/2$ であるか否かを判定しているが、例えば $1/3$ 、 $1/4$ 、 $3/5$ などなんでもよい。領域を効率よく利用することを考慮すれば、この値は0以上から約 $1/2$ 以下であることが望ましい。

(2) ステップ403においてGC起動部11は、許可フラグが0であると判定した場合、さらに領域監視部10に保持される未使用領域のアドレスからヒープ領域に示す未使用領域の大きさの割合が $1/4$ 以下(この値はステップ401において設定されている値より小さければよい)であるかを判定し、 $1/4$ 以下と判定した場合にはGC処理部12にGCを実行させ、 $1/4$ 以下ではないと判定した場合にはこのフローチャートの処理を終了するように構成してもよい。

(3) 時間制限命令の制限時間を設定する代わりに、1回のGCにおいて確保すべき未使用領域の容量を設定するようにしてもよい。すなわち、1回のGCにおいて、使用済み領域から未使用領域に戻した容量が所定量(例えばヒープ領域の $1/4$ 量)に達した場合にそのGCを終了する、というように構成してもよい。

(4) 図1に示すプログラム実行装置1の構成は、カウンタ更新部8とカウンタ値保持部9とを除いた構成にして、プログラマが記述できる制限情報は時間制限命令を除く禁止命令と許可命令のみにしてもよい。

【0034】

【発明の効果】本発明のプログラム実行装置は、プログラムの実行中に、動的記憶領域に含まれる使用済み領域を回収して未使用領域にするガベージコレクションを行うプログラム実行装置であって、動的記憶領域における未使用領域の状況を監視する監視手段と、ガベージコレクションの実行を制限する制限情報を保持する保持手段と、前記状況と前記制限情報とに応じてガベージコレクションを実行する実行手段とを備える。

【0035】この構成によれば、実行手段が未使用領域の状況と制限情報とに応じてガベージコレクションを実行するので、プログラマはガベージコレクションを行わせるために逐一解放命令を記述する必要がなく、プログラマにかかる負担が軽減されるという効果がある。また実行手段によるガベージコレクションは制限情報によって制限されるので、従来のようにタイミングと実行時間が予測不可能であったガベージコレクタによるガベージコレクションと比べると、一定の時間内に所定の処理を終了することが保証されるので、本プログラム実行装置はリアルタイム処理の実行により適している。

【0036】また前記制限情報は、ガベージコレクションの実行の禁止又は許可を規定する情報であり、前記監視手段は、動的記憶領域中に占める未使用領域の割合を

監視し、前記実行手段は、前記割合が所定値以下になった場合に、前記保持手段に保持されている制限情報が前記許可を規定する情報であるときガーベジコレクションを実行するよう構成される。

【0037】この構成によれば、プログラムにおいてリアルタイム処理が要求される場合にはガーベジコレクションを禁止し、要求されない場合にはガーベジコレクションを許可することができる。また、実行手段は、プログラム上の前記許可を規定する情報が有効な期間において、前記割合が所定値以下になった場合にガーベジコレクションを実行するので、プログラマは、逐一ガーベジコレクションの実行を指示する必要がなくなり、プログラマにかかる負担が軽減される。

【0038】また前記許可を規定する情報は、1回のガーベジコレクションの実行時間の上限を規定することを特徴とする。これにより実行手段が1回に実行するガーベジコレクションの実行時間を制限することができるので、プログラムの処理に要する時間を予測することができる。よって本プログラム実行装置は、予測性が要件として要求されるリアルタイム処理プログラムを実行することができる。

【0039】また前記許可を規定する情報は、1回のガーベジコレクションにおいて確保すべき未使用領域の容量を規定することを特徴とする。これにより実行手段は、使用済み領域が前記容量より大きい場合においても、容量分のガーベジコレクションを行ったら中断するので、1回のガーベジコレクションにかかる時間を容量分に抑えることができる。よって本プログラム実行装置は、プログラムの処理を期待される時間内に完了させることができるので、リアルタイム処理の実行に適している。

【0040】また前記制限情報は、プログラムを構成する各部分におけるリアルタイム処理の必要性に応じて各部分の先頭に記述され、前記保持手段は、前記プログラムより前記制御情報が読み出された時点からそれを保持するよう構成される。これによりプログラマは、プログラム実行装置にリアルタイム処理を行わせる部分の先頭に前記禁止を規定する情報を記述し、リアルタイム処理を行わせない部分の先頭に前記許可を規定する情報を記述するだけで、その記述以降の実行手段によるガーベジコレクションの実行を制限することができるので、プログラマは逐一ガーベジコレクションの実行をプログラム上で記述する必要がなくなるという効果がある。

【図面の簡単な説明】

【図1】プログラム実行装置の構成を示す図である。

【図2】制限情報が記述されたプログラムの一例を示す。

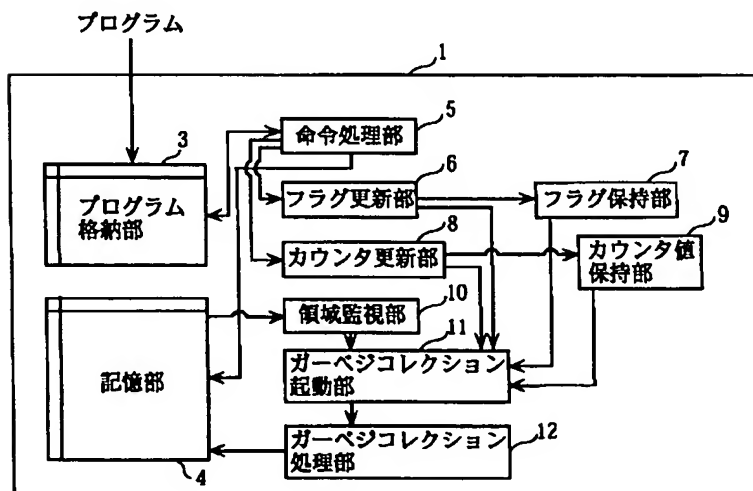
【図3】命令処理部5、フラグ更新部6、カウンタ更新部8を中心とする処理手順のフローチャートである。

【図4】GC起動部11によるGC制御処理の手順を示すフローチャートである。

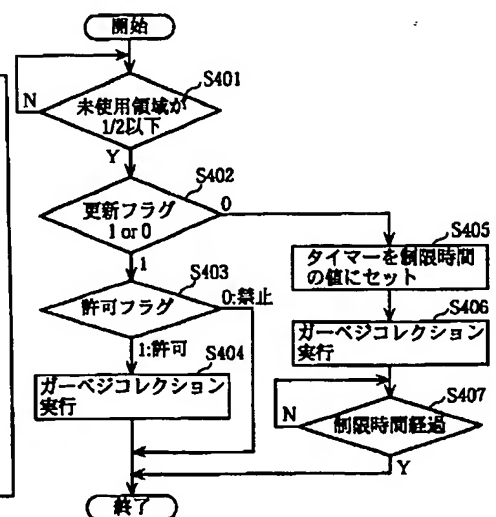
【符号の説明】

- 3 プログラム格納部
- 4 記憶部
- 5 命令処理部
- 6 フラグ更新部
- 7 フラグ保持部
- 8 カウンタ更新部
- 9 カウンタ値保持部
- 10 領域監視部
- 11 GC起動部
- 12 GC処理部

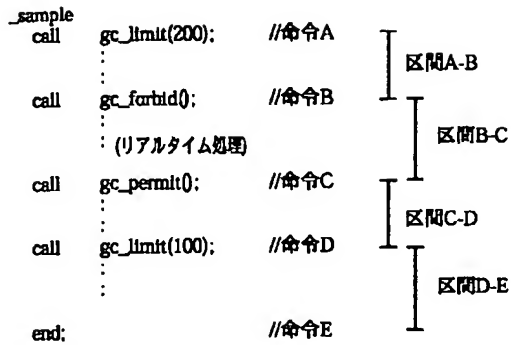
【図1】



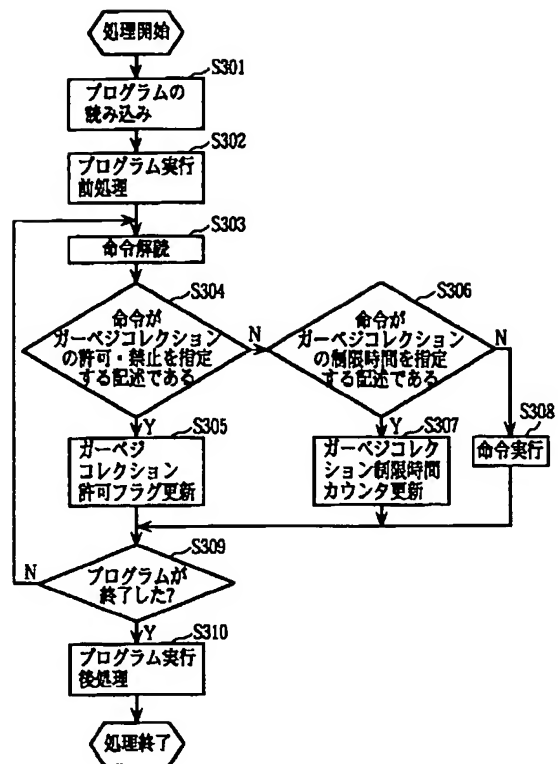
【図4】



【図2】



【図3】



フロントページの続き

(72)発明者 春名 修介
大阪府門真市大字門真1006番地 松下電器
産業株式会社内

Fターム(参考) 5B060 AA10 AA14